

BEST AVAILABLE COPY**In the Claims:**

Please cancel claims 7-17. Please amend claims 1 and 4. Please add new claims 18-33.

The claims are as follows:

1. (Currently amended) A computer software system comprising:

a view sub-system including presentation objects which provide a user interface;

a business logic sub-system including business object implementation objects which hold business data objects and implement business functions;

[[a]] first and second handler sub-systems, each said handler sub-system including controller objects which control a sequence of actions by the business logic sub-system in a use case, in response to an event triggered by the view sub-system; and

a view context sub-system including at least one a view context object which is arranged to capture input and output data which populates the presentation objects of the view sub-system, wherein the software system is configured to have the view context object:

passed from the view sub-system to the first handler sub-system,

passed from the first handler sub-system to the second handler sub-system such

that the view context object is updated by at least one of the first handler sub-system and the second handler sub-system, and

passed after being updated from the second handler sub-system to the view sub-system and subsequently used by the view sub-system to refresh a view.

BEST AVAILABLE COPY

2. (Cancelled)

3. (Original) The computer software system of claim 1, wherein:

the view context sub-system also comprises data interfaces for the business logic sub-system.

4. (Currently amended) The computer software system of claim 1, wherein[:]

~~the context objects included in the view context sub-system are updated when input is entered into the view sub-system by a user, and~~

~~the view context data objects are is updated by the at least one of the first handler sub-system and the second handler sub-system whenever business logic is executed on any of the view context object[s].~~

5. (Original) The computer software system of claim 1 wherein:

the view sub-system refreshes the presentation objects with the input and output data from the view context sub-system.

6. (Original) The computer software system of claim 1, wherein:

the view context sub-system is represented in a platform-independent format.

7-17. (Cancelled)

BEST AVAILABLE COPY

18. (New) The computer software system of claim 1, wherein the view context object is updated by both the first handler sub-system and the second handler sub-system.

19. (New) The computer software system of claim 1, wherein the updated view context object is passed from the second handler sub-system to the view sub-system by:

being passed from the second handler sub-system to the first handler sub-system, and subsequently

being passed from the first handler sub-system to the view sub-system.

20. (New) A method of passing data within an object oriented software environment, said method comprising the computer-implemented steps of:

providing a view sub-system including presentation objects which provide a user interface;

providing a business logic sub-system including business object implementation objects

which hold business data objects and implement business functions;

providing first and second handler sub-systems, each said handler sub-system including controller objects which control a sequence of actions by the business logic sub-system in a use case, in response to an event triggered by the view sub-system;

providing a view context sub-system including a view context object which is arranged to capture input and output data which populates the presentation objects of the view sub-system, wherein the software system is configured to have the view context object;

passing the view context object from the view sub-system to the first handler sub-system;

passing the view context object from the first handler sub-system to the second handler

sub-system such that the view context object is updated by at least one of the first handler sub-system and the second handler sub-system; and

passing the updated view context object from the second handler sub-system to the view sub-system; and

refreshing a view by the view sub-system, said refreshing using the updated view context object.

21. (New) The method of claim 20, wherein:

the view context sub-system also comprises data interfaces for the business logic sub-system.

22. (New) The method of claim 20, wherein the view context data object is updated by the at least one of the first handler sub-system and the second handler sub-system whenever business logic is executed on the view context object.

23. (New) The method of claim 20, wherein:

the view sub-system refreshes the presentation objects with the input and output data from the view context sub-system.

24. (New) The method of claim 20, wherein the view context sub-system is represented in a platform-independent format.

BEST AVAILABLE COPY

25. (New) The method of claim 20, wherein the view context object is updated by both the first handler sub-system and the second handler sub-system.

26. (New) The method of claim 20, wherein the updated view context object is passed from the second handler sub-system to the view sub-system by:

passing the updated view context object from the second handler sub-system to the first handler sub-system; and subsequently

passing the updated view context object from the first handler sub-system to the view sub-system.

27. (New) A computer program adapted to be executed on a processor of a computer system to implement a method of passing data within an object oriented software environment, said method comprising:

providing a view sub-system including presentation objects which provide a user interface;

providing a business logic sub-system including business object implementation objects which hold business data objects and implement business functions;

providing first and second handler sub-systems, each said handler sub-system including controller objects which control a sequence of actions by the business logic sub-system in a use case, in response to an event triggered by the view sub-system;

providing a view context sub-system including a view context object which is arranged to capture input and output data which populates the presentation objects of the view sub-system, wherein the software system is configured to have the view context object;

BEST AVAILABLE COPY

passing the view context object from the view sub-system to the first handler sub-system;
passing the view context object from the first handler sub-system to the second handler sub-system such that the view context object is updated by at least one of the first handler sub-system and the second handler sub-system; and
passing the updated view context object from the second handler sub-system to the view sub-system; and
refreshing a view by the view sub-system, said refreshing using the updated view context object.

28. (New) The computer program of claim 27, wherein:

the view context sub-system also comprises data interfaces for the business logic sub-system.

29. (New) The computer program of claim 27, wherein the view context data object is updated by the at least one of the first handler sub-system and the second handler sub-system whenever business logic is executed on the view context object.

30. (New) The computer program of claim 27, wherein:

the view sub-system refreshes the presentation objects with the input and output data from the view context sub-system.

31. (New) The computer program of claim 27, wherein the view context sub-system is represented

in a platform-independent format.

BEST AVAILABLE COPY

32. (New) The computer program of claim 27, wherein the view context object is updated by both the first handler sub-system and the second handler sub-system.

33. (New) The computer program of claim 27, wherein the updated view context object is passed from the second handler sub-system to the view sub-system by:

passing the updated view context object from the second handler sub-system to the first handler sub-system; and subsequently
passing the updated view context object from the first handler sub-system to the view sub-system.